REMARKS

Claims 1-17 are pending.

Claims 1-17 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,721,727 (Chau et al.). Reconsideration is respectfully requested.

Applicants wish to thank Examiner Bell for taking the time to conduct a telephonic interview on November 29, 2006. During the interview, the foregoing claim amendments and the following arguments were discussed.

The present invention is directed to a method that enables XML data to be stored in a field of a user defined type, ***while other fields of the user defined type can store other types of data***. For example, as illustrated in Figure 5, a user defined type called "Employee" can be created having fields called fName, lName, Salary, Age and Resume. An instance of the Employee user defined type can thus be used to store an employee's name, salary, age and resume. The fName, lName, Salary and Age fields are defined in this example as having the traditional string, floating point (*i.e.*, "double") and integer data types, respectively, whereas the Resume field is defined to hold XML data.

The ability to store XML data in a field of a user defined type is achieved, at least in part, by first defining a new XML data type as a class in managed code. In the present embodiment, this new class is called "SqlXml". *Specification, ¶¶* 46-50. Then, a user defined type can be created in which at least one of the fields of the user defined type is defined as having the new XML data type, *i.e.,* the field is defined as an instance of the SqlXml class. Referring again to the example illustrated in Figure 5, the "Employee" user defined type has a field called "Resume" that is defined as having the new "SqlXml" type. That field of the user defined type can thus be used to store an XML document, *e.g.,* an employee's resume. Thus, the present invention enables XML data to be stored in a field of an instance of a user defined type, ***while other fields of the user-defined type store other types of data***.

Applicants have amended independent claims 1, 7 and 12 to emphasize this novel aspect of the invention. For example, amended claim 1 now recites:

> defining ***at least one of the plurality of fields*** of the user defined type as having the XML data type, and defining ***at least***

> ***one other of said plurality of fields*** as having a ***different*** data type;

Claims 7 and 12 recite similar features. Chau does not teach or suggest defining at least one field of a user defined type as having an XML data type, while defining other fields of the user defined type as having other data types, as claimed.

Anticipation under 35 U.S.C. § 102(e) requires that each and every limitation of a claim be met in a single reference. The Office Action suggests that Chau's "DAD" is equivalent to the recitation in claims 1, 7 and 12 of "a user defined type ... defined by a class in managed code and compris[ing] a plurality of fields, each field having a respective data type." Applicants respectfully disagree.

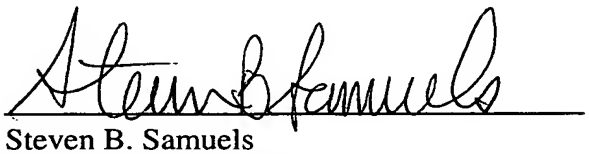
As those of ordinary skill in the art know, a "class" is a programming tool used in object-oriented programming to define a collection of encapsulated variables and methods (sometimes called "behaviors"). Once defined, a class can be "instantiated" to create an instance (*i.e.*, an object) of that class. A "user defined type" that is defined by a class in managed code can thus be instantiated as an object (*i.e.*, an instance) having that user defined type (*e.g.*, an instance of the Employee user defined type shown in Figure 5). Independent claims 1, 7 and 12 have been amended at the Examiner's suggestion to make this more clear.

The "XML System" described by Chau is designed to allow a user to store XML documents as column data in a relational database table or to transform between XML documents and data in existing relational database tables. Chau, col. 7, ll. 43-45. The "DAD" described in Chau is a "Document Access Definition" that is used to define how an XML document is to be stored to or created from the columns of one or more relational database tables. Chau, col. 7, ll. 49-51. As Chau further explains, "[t]he DAD itself *is an XML formatted document.*" Col. 7, ln. 52 (emphasis added). Thus, a DAD is ***not*** a "user defined type," nor is a DAD defined by "a class in managed code," as recited in claims 1, 7 and 12.

Because of the differences between a Document Access Definition (DAD) in Chau and a "user defined type" as recited in claims 1, 7 and 12, the use of DADs in the system of Chau cannot be said to anticipate those claims. Nowhere does Chau describe the ability to create an object that is an instance of a user defined type having a plurality of fields, where at least one field is defined as having an XML data type while other fields of the user defined type are defined as having other data types, as claimed by applicants. For this reason,

applicants respectfully submit that claims 1, 7 and 12 of the present application patentably define over Chau and any other cited art of record. Inasmuch as the other claims all depend either directly or indirectly from one of these independent claims, Applicants submit that they too patentably define over the art of record. Reconsideration of the Section 102(e) rejection of claims 1-17 is therefore respectfully requested.

Date: December 19, 2006



Steven B. Samuels
Registration No. 37,711

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439